Digitally-controlled PC-interfaced Boost Converter for Educational Purposes

Ljusev, Petar; Andersen, Michael A. E.

Published in:
Ninth IEEE Workshop on Computers in Power Electronics COMPEL 2004

Link to article, DOI:
10.1109/CIPE.2004.1428156

Publication date:
2004

Document Version
Publisher's PDF, also known as Version of record

Digitally-controlled PC-interfaced Boost Converter for Educational Purposes

Petar Ljušev, Michael A.E. Andersen
Ørsted • DTU, Automation
Technical University of Denmark
Elektrovej DTU Building 325
Kgs. Lyngby DK-2800, Denmark
Telephone: +45 4525 3486, Fax: +45 4588 6111
Email: pl@oersted.dtu.dk, ma@oersted.dtu.dk

Abstract— **This paper describes implementation of a simple digital PID control algorithm for a boost converter using a cheap fixed-point 8-bit microcontroller. Serial communication to a PC server application is established for easier downloading of compensator parameters and current and voltage waveform acquisition. At the end, client application is presented which uses TCP/IP connection for operating the digitally controlled boost converter over Internet. The aim of this cheap and flexible PC-interfaced boost converter bench is predominantly educational, to allow students to synthesize different digital controllers and compare their performance.**

I. Introduction

The field of DC-DC converters and their control has been up to now extensively and thoroughly investigated and elaborated [1], so that there are almost no unresolved issues. While the analog control algorithms in general perform very good, implementing digital control of DC-DC converters is still tedious. Analog current mode PWM controllers are common on the market and their price is affordable, while implementing the control in a microcontroller with the associated analog measuring and conditioning circuitry can be quite expensive, not mentioning the time spent to develop the code and debug it. When implementing the control in a cheap fixed-point 8-bit microcontroller with built-in A/D converters and PWM counters, many internal restrictions (low clock speed, reduced instruction set with only 8-bit register manipulation, no instruction pipelining and parallelism, no hardware multiplication, possible limit cycling due to insufficient PWM modulator resolution) can substantially limit the performance and control bandwidth. Some ways to alleviate these problems have been already addressed [2], [3], [4].

However, once the analog controller has been designed and hard-wired, the product is self contained and no further changes are possible. Communication with the power supply is usually restricted to "Power good" signals, which cannot give enough insight in the power supply condition or help troubleshooting. On the other hand, digital based systems can offer: flexibility in updating the control program and controller parameters, lower noise sensitivity, programmability without external components and easy communication with the host system using some protocol (RS232, RS485, SPI, I^2C etc).

Due to these capabilities and obvious advantages of the digitally controlled converters, an aim was set for the project to: develop a simple DC-DC converter controlled by a microcontroller using a PID control algorithm; use serial communication RS232 to communicate with a PC, where the measured quantities will be processed further and presented in a meaningful way; download new PID compensator gains to the microcontroller using a graphical user interface (GUI) of the PC application; help students in choosing the stabilizing gains of the PID compensator to improve the power supply dynamics and reject disturbances; allow the server application connected to the test bench to communicate with similar client applications distributed over the Internet using TCP/IP protocol, so that the interested students can access the test bench and learn by making experiments on a real power supply from home.

The project was done during the three weeks course period in June-July 2003, as a part of the "Advanced Power Electronics" course at the Technical University of Denmark (DTU) in Kgs. Lyngby. Due to the narrow time frame, it was decided that more attention will be payed to achieving the educational objective i.e. developing a power electronics system that is easy to control through a PC application and the specially designed GUI, building a bridge from the custom application to MATLAB to alleviate the control design and opening a gateway from the test bench to the internet to let students access it from a distance. The complexity and performance of the power converter itself was put in second plan, although its selection was done with regard to the educational goals and the student abilities this test bench should develop.

For this "educational purpose" project, Microchip microcontroller PIC16F877 [5] with the In-Circuit Debugger (MPLAB ICD) was already available. Although the processor core is running at "moderate" clock rate of 20 Mhz and has no hardware multiplier, it has plenty of peripheral circuitry like A/D converters, timers, counters, PWM modules, USART and plenty of interrupt sources that made the implementation much easier. Additional advantage is the relatively small instruction set of 35 single words to learn as well as plenty of available resources and math routine libraries for the aforementioned family of microcontrollers.

0-7803-8502-0/04/$20.00 ©2004 IEEE.

II. BOOST CONVERTER

When having the practical and educational considerations in mind, the boost converter operating in continuous conduction mode (CCM) was a highly appealing DC-DC converter to select. It was concluded that the bench would help students clarify the concept of stepping-up the DC input voltage and show a power electronics structure with complex control-to-output transfer function $G_{vd}(s)$ (two poles and one right-half plane zero). Apart from this educational objective, the construction itself was alleviated due to the possibility of using an N-type MOSFET referenced to ground, which necessitates only a simple low-side driver.

The boost DC-DC converter design specifications were chosen as follows: input voltage $V_g = 5$ V, output voltage $V = 12$ V, output power $P_{out} = 12$ W, maximum voltage and current ripple: $\Delta v_{max} = 1\%\ V_{out}$ and $\Delta i_{max} = 0.2$ A. With regard to the capabilities of the PIC16F877 microcontroller, it was obvious that enough time should be allocated for the control algorithm to execute. This means that the switching frequency should be low, but certainly out of the audio band. Therefore, the switching frequency was chosen to be $f_s = 20$ kHz. These specifications were used to select the output filter and semiconductor components.

In order to be able to measure all the desired converter quantities: converter input and output voltage V_{in} and V_{out} and input and output current I_{in} and I_{out}, the boost converter was supplied with resistive voltage dividers and current sense resistors.

Regarding the converter control, it was concluded that implementing digital control of only the output voltage will suffice. Current programmed control, although very desirable and popular in power supplies, would further complicate both the design and the program code and will necessitate use of separate analog comparators or more complex microcontrollers with built-in comparators to achieve high speed of the inner loop.

III. INTERFACING THE MICROCONTROLLER TO THE POWER CONVERTER

When interfacing a power converter to a digital controller, properly conditioned analog to digital (A/D) conversion is of profound importance. The project proposal demanded that the microcontroller is used not only for control, but also for data acquisition of the input/output voltages/ currents. With four quantities to be measured, the acquisition time was requested to be rather short and certainly not longer than few microseconds. This was not possible with the internal 10-bit A/D converters of PIC16F877 which have a total conversion time around 30 μs, so an external 10-bit A/D converter AD7470 with a parallel output was used, allowing for an order of magnitude faster acquisitions.

In order to provide for proper A/D conversion of measured voltages and currents, A/D converter was preceded by third-order active anti-aliasing filters, designed to give less than 1 LSB error at $f_s/2 = 10$ kHz, i.e. provide -60 dB of attenuation, as well as sample-and-hold and multiplexer circuitry. Active filters provide also a lower output impedances which are necessary for faster settling of the sampling section of the sample-and-hold circuit, multiplexer and the A/D converter. To finish the subject of interfacing, a dedicated gate-driver chip was used for switching the logic-level power MOSFET.

IV. DIGITAL CONTROL OF THE CCM BOOST CONVERTER

Students usually find the analog control synthesis easier, with the Bode-diagram loop shaping and phase-margin specifications being their favorite tool. Therefore, the design was performed entirely in the analogue domain, by mapping all the digital transfer functions into the latter.

The implemented control block diagram is given in Fig. 1.

Control-to-output transfer function $G_{vd}(s)$ of the continuous conduction mode boost converter is [1]:

$$G_{vd}(s) = \frac{V}{D'} \cdot \frac{1 - \frac{sL}{D'^2R}}{1 + s\frac{L}{D'^2R} + s^2\frac{LC}{D'^2}} \tag{1}$$

where V is the output voltage, $D' = 1 - D$ is the complement of the duty-cycle D, L is the filter inductance, C is the filter capacitance and R is the load resistance. One of the factors that strongly limits the available performance from a CCM boost converter is the right half-plane zero at the angular frequency of $\omega_{RHPZ} = D'^2R/L$, dropping the maximum control bandwidth usually to around $\omega_{RHPZ}/3$ [6].

The transfer function of the implemented third-order anti-aliasing filter is:

$$H_v(s) = \frac{v_o(s)}{v(s)} = \frac{R_2}{R_1 + R_2} \cdot \frac{1}{a_0 s^3 + a_1 s^2 + as + 1} \tag{2}$$

where R_1, R_2 form a scaling-down resistive divider network and a_0, a_1, a_2 are the anti-aliasing filter coefficients.

Due to the short acquisition and conversion time of the A/D converter, it is assumed that it does not introduce any dynamics, so the transfer function is constant $K_{AD} = 1023/2.5$ (code $1023 \equiv V_{ref} = 2.5\ V$).

The continuous-time PID compensator is given by:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \tag{3}$$

where $e(t)$ is the error signal and $u(t)$ is the compensator output.

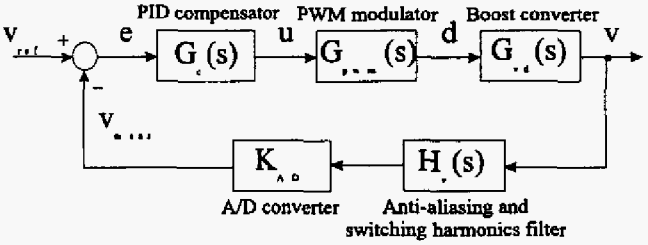

Fig. 1. Digital control system of the boost converter

The discrete version of the PID compensator in (3) is:

$$u(kT) = K_p e(kT) + K_i' \sum_{n=0}^{k} e(nT) + K_d' \left[e(kT) - e((k-1)T) \right] \quad (4)$$

where $K_i' = K_i T$ is the discrete integral gain, $K_d' = K_d / T$ is the discrete derivative gain, and T is integration/derivation time period ($T = nT_s$ where T_s is the sampling period and n represents the number of sampling periods T_s used to calculate the PID terms).

Using the Laplace transform on (3) the ideal PID compensator transfer function in the s-domain is obtained:

$$G_c(s) = K_p + \frac{K_i}{s} + sK_d = K_i \frac{1 + s\frac{K_p}{K_i} + s^2 \frac{K_d}{K_i}}{s} \quad (5)$$

Moving to frequency domain $s \rightarrow j\omega$, (5) becomes:

$$G_c(j\omega) = K_p + \frac{K_i}{j\omega} + j\omega K_d = K_p + j\left(\omega K_d - \frac{K_i}{\omega} \right) \quad (6)$$

The magnitude $|G_c(j\omega)|$ of (6) in dB is:

$$A = 20 \log |G_c(j\omega)| = 10 \log \left[K_p^2 + \left(\omega K_d - \frac{K_i}{\omega} \right)^2 \right] \quad (7)$$

while the phase $\angle G_c(j\omega)$ is:

$$\phi(\omega) = \arctan \frac{\omega K_d - \frac{K_i}{\omega}}{K_p} = \arctan \frac{\omega^2 K_d - K_i}{\omega K_p} \quad (8)$$

Equations (7) and (8) can be used to calculate K_p, K_i and K_d to achieve certain magnitude A and phase ϕ at angular frequency of ω. From (8), proportional gain is found to be:

$$K_p = \frac{\omega K_d - \frac{K_i}{\omega}}{\tan \phi} = \frac{\omega T K_d' - \frac{K_i'}{\omega T}}{\tan \phi} \quad (9)$$

When the proportional gain K_p from (9) is substituted into (7), the derivative gain K_d becomes:

$$K_d = \pm \frac{1}{\omega} \sqrt{\frac{10^{\frac{A}{10}} \tan^2 \phi}{1 + \tan^2 \phi}} + \frac{K_i}{\omega^2} \quad (10)$$

or in terms of discrete derivative and integral gain K_d' and K_i':

$$K_d' = \pm \frac{1}{\omega T} \sqrt{\frac{10^{\frac{A}{10}} \tan^2 \phi}{1 + \tan^2 \phi}} + \frac{K_i'}{\omega^2 T^2} \quad (11)$$

Once the desired compensator magnitude A and phase ϕ at the angular frequency ω are known, K_p and K_d' are determined by iterating through all achievable discrete integral gains K_i' using equations (9), (11). The aforementioned method can be used to determine the stabilizing compensator which results in particular crossover frequency ω_c, but there is no guarantee neither for sufficient gain throughout the control bandwidth nor for stability, since the zero gain crossing of the transfer function is unique only if the system transfer function is monotonous i.e. without any resonant peaking.

The PWM counter resolution, with regard to the chosen switching frequency of 20 kHz and the oscillator frequency of 20 MHz is almost 10 bits, which barely avoids limit cycling [2]. Digital code of 1023 corresponds to a duty cycle of $D = 1$, so ideally the transfer function of the PWM module is equal to their ratio. However, the calculated duty cycle from the PID compensator takes effect at the end of the PWM period to allow glitch-free PWM operation. Thus a delay of one switching period T_s is introduced in the PWM modulator. Unfortunately, the PWM modulator is not the only source of delays. After the PID algorithm was implemented in its complete form by extensive use of software multiplication, it was found out that calculation of each of the PID terms resulted in one additional switching period delay T_s. For example, with all terms calculated (three multiplications/divisions and many additions/subtractions) it was concluded that a maximum of 3 switching periods was needed to calculate the new duty cycle, based on values measured 3+1=4 switching periods behind, one being for the delay introduced in the PWM. In order to do the control synthesis, all of these delays were "dumped" into the PWM modulator transfer function to yield:

$$G_{pwm}(s) = \frac{1}{1023} e^{-snT_s} \; ; \; G_{pwm}(z) = \frac{1}{1023} \cdot z^{-n} \quad (12)$$

where $n = 1..4$ is the total number of T_s delays from the moment the control algorithm started calculating till the moment the new duty cycle is loaded into the PWM modulator.

V. Microcontroller Program Implementation

Microcontroller program was divided into three major parts: initialization, main program and interrupt service routine.

Initialization part of the code is the one which is executed at start-up and where all the initial settings (ex. defining port direction, PWM period, interrupt configuration, reset of PID registers, USART speed etc.) are performed.

Main program was made very small and was filled with code for serial communication to the PC server application. This is logical, since the measured data transfer to the server application is of lower priority when compared to the control routine, and therefore can be interrupted whenever necessary.

The interrupt service routine is where most of the "meat" of the program is situated. There are several interrupts that have been used for proper operation of the converter. The "overcurrent limit reached" is the most significant one, which happens when the active switch current is over the limit and therefore it is not masked at all. The USART interrupt service routine, which has a slightly lower priority, only receives the data from PC for processing, since the transmission of data is left to the main program. Reason for the high priority of the latter is that no data flow control was implemented and PIC16F877 has only a 2-level deep receive stack, so the incoming data should be read as soon as possible to avoid any overrun condition. Timer interrupt is scheduled to occur each milisecond, taking the last measured values of the input/output voltages/currents and loading them into separate registers to be available for the next serial transmission to the PC server application. In this way, control algorithm can continue sampling and storing new values for V_{in}, I_{in}, V_{out}, I_{out}, but only those captured at 1 ms instants are sent for further

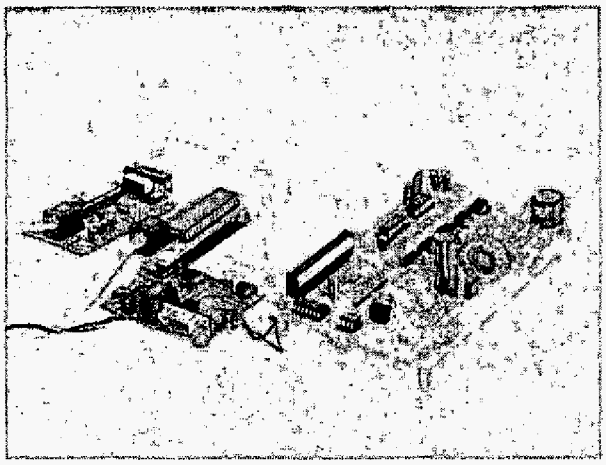

Fig. 2. Photo of the power electronics bench

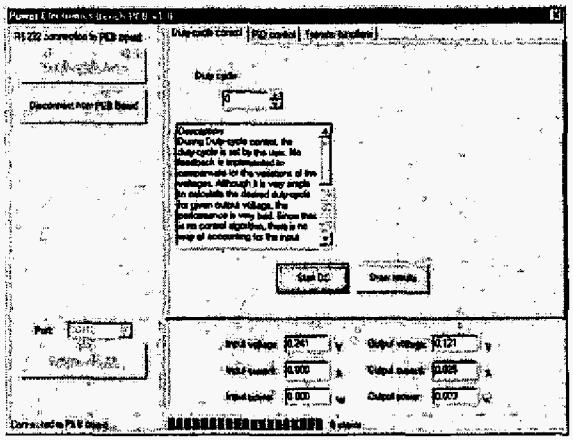

Fig. 3. PEBv1.0 - Duty-cycle control page

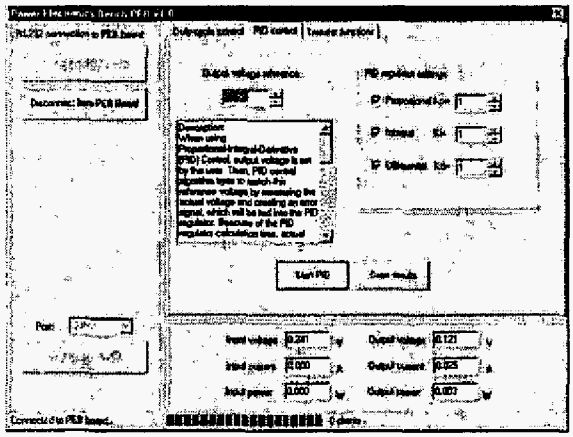

Fig. 4. PEBv1.0 - PID control page

processing to the PC server application, due to the speed limitations of the serial RS232 link. The last, the longest and the lowest priority interrupt service routine is the one for the PWM interrupt. It is used to perform all of the steps, which are important for the boost converter control, like sampling of the measured quantities and calculating the output of the PID compensator.

In order to keep the implementation of the PID compensator simple, it was desired to use 10-bit integer unsigned values everywhere in the calculations and keep the signs in a separate registers. However, the use of slower double precision (16-bit) math routines could not be avoided. In order to allow discrete gains both higher and lower than 1 as requested by the control synthesis calculations, both double precision multiplication and double precision division were implemented. Each of the PID terms in (4) were separately calculated and limited, as well as their corresponding sum.

The photo of the power electronics bench with the boost converter on the right, microcontroller in the middle and the MPLAB In-Circuit Debugger (ICD) on the left is shown in Fig. 2.

VI. Server and Client PC Applications

The server and client PC applications "Power Electronics Bench PEB v1.0" (PEBv1.0) are programmed in Delphi [7] and are using the RS232 for communicating data with the microcontroller. Both the server and client applications are the same, but the former is connected directly to the power electronics bench, while the latter is used for connecting to the server application over the internet. The instantaneous values of the voltages, currents and powers are calculated in the PC application, while transfer functions and waveforms are graphically represented in MATLAB using "Object Linking and Embedding" (OLE) for data transfer between the applications.

The PC application has three main windows: duty-cycle control (open-loop) shown in Fig. 3 , PID control (closed-loop) shown in Fig. 4 and transfer functions shown in Fig. 5. The last one is used to help students grasp the complex control structure of the overall test bench, as well as to plot the transfer functions and choose the approximately correct PID parameters through the GUI in Fig. 6. The latter is achieved using equations (9), (11) and the students can choose one of the proposed parameters set which is giving the desired crossover frequency and phase margin. In the cases where the specified phase margin cannot be achieved due to the limitation of the phase shift inserted by the PID compensator ($-90° < \phi_{COMP} < 90°$) program doesn't return any set of parameters and informs about the inability to properly compensate the open loop transfer function. In order to ease the selection of the compensator parameters, the list is limited just to those parameter combinations which are within the range of the microcontroller word length representation.

It should be noted however, that the accuracy of the aforementioned helping tool is limited due to the resonant peaking of the second-order converter transfer function, which often causes the resultant crossover point to be non-unique.

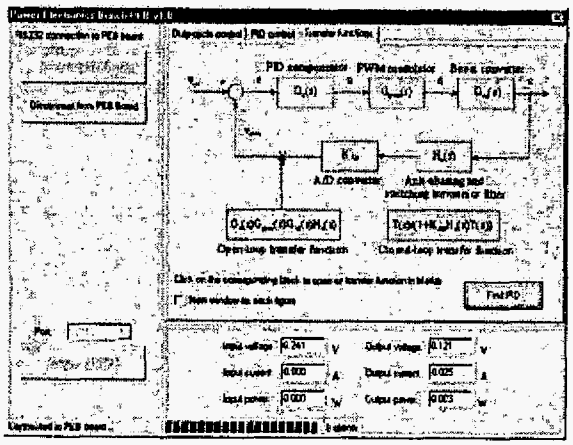

Fig. 5. PEBv1.0 - Transfer functions page

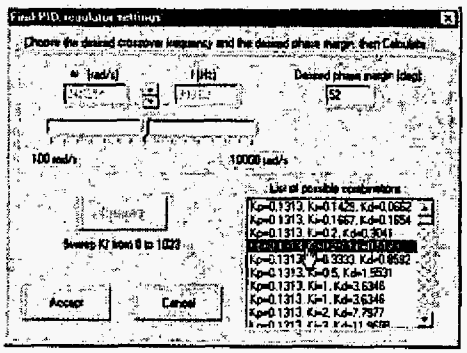

Fig. 6. PEBv1.0 - "Find PID compensator settings" form

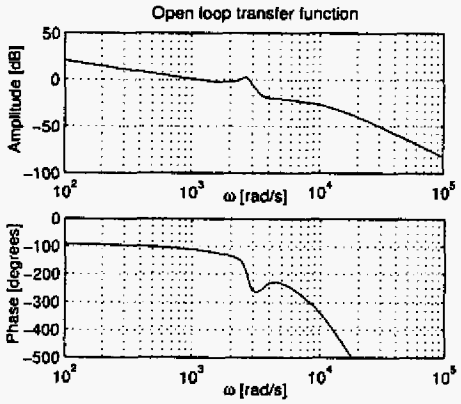

Fig. 7. Open-loop transfer function

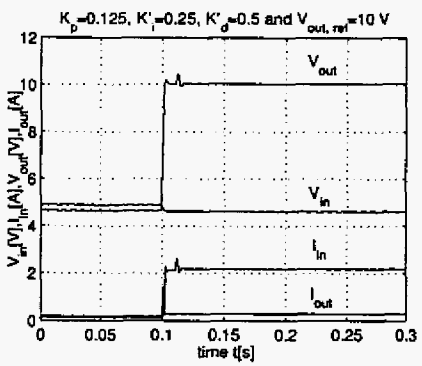

Fig. 8. Experimental waveforms with step in the reference at t=0.1 s

VII. Example

In this example, a PID compensator is designed which results in 52° degrees of phase margin of the open loop transfer function at angular frequency of $\omega_c = 2450$ rad/s ($f = 390$ Hz). Due to the flexible digital control, the boost converter can give a wide range of stable output voltages higher than Vg and limited solely by the parasitics of the converter itself. In order to test this capability, the output voltage is chosen to be $V_{out} = 10$ V. After entering the predefined values in the form in Fig. 6, pressing the calculate button will give a plenty of choices. Although all of the proposed compensator parameter combinations lead to zero gain of the resulting open loop transfer function at the desired crossover frequency ω_c, the overall shape of the transfer function differs remarkably, rendering many of the combinations inappropriate. Choosing a suitable combination with enough integral gain, for ex. $K_p = 0.1313$, $K_i = 0.25$, $K_d = 0.5123$ ($K_p = 0.125$, $K_i = 0.25$, $K_d = 0.5$ after rounding to microcontroller precision), gives the open-loop transfer function in Fig. 7 and the step response in Fig. 8.

VIII. Conclusion

In this paper a digitally controlled power electronics bench with a boost converter, PC interface and internet connection was presented. It is developed to serve the purpose of a teaching aid for the power electronics students in class or for distant learning. Although the educational goals and the impact on students were put in front of the converter performance, it is believed that this bench successfully paves the way to similar projects providing excellence in power electronics education at the University.

References

[1] R. W. Erickson and D. Maksimović, *Fundamentals of Power Electronics*. Kluwer Academic Publishers, 2001, second edition.

[2] A. Prodic, D. Maksimovic, and R. Erickson, "Design and implementation of a digital pwm controller for a high-frequency switching dc-dc power converter," in *Proc. The 27th Annual Conference of the IEEE Industrial Electronics Society, 2001. IECON '01.*, vol. 2, Denver, Colorado, USA, November 2001, pp. 893–898.

[3] A. Prodic and D. Maksimovic, "Design of a digital pid regulator based on look-up tables for control of high-frequency dc-dc converters," in *Proc. 8th IEEE Workshop on Computers in Power Electronics, 2002.*, Mayaguez, Puerto Rico, June 2002, pp. 18–22.

[4] A. Peterchev and S. Sanders, "Quantization resolution and limit cycling in digitally controlled pwm converters," *IEEE Trans. Power Electron.*, vol. 18, no. 12, pp. 301–308, 2003.

[5] *PIC16F87X Datasheet - 28/40-Pin 8-bit CMOS FLASH Microcontrollers*, Microchip Inc., 2001, dS30292C.

[6] J. Lloyd H. Dixon, "Control loop cookbook," *Unitrode power supply design seminar SEM1100*, pp. 5.1–5.26, 1996.

[7] "Delphi v.7 architect," borland. [Online]. Available: www.borland.com/delphi